International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 2, Apr 2014, 285-290

SPEECH RECOGNITION OF HINDI PHENOMES USING MFCC AND BHATTACHARRYA HISTOGRAM DISTANCE

SANDEEP KAUR[1], MEENAKSHI SHARMA[2] & SUKHBEER SINGH[3]

[1]M.Tech Student, Department of CSE, Sri Sai College of Engineering, Pathankot, Punjab, India

[2]Department of Head, Sri Sai College of Engineering, Pathankot, Punjab, India

[3]Assistant Professor, Sri Sai College of Engineering, Pathankot, Punjab, India

ABSTRACT

This paper describes an algorithm that takes advantage of the distance measures for finding similarity between the histogram profiles of the feature matrix made of audio signals (Hindi Phenomes). The results obtained with Swaranjali for tests conducted on a vocabulary of Hindi digits of different speaker. Many researchers have used the root mean square (rms), log spectral distance, cepstral distance, likelihood ratio (minimum residual principle or delta coding (DELCO) algorithm), and a cosh measure (based upon two non symmetrical likelihood ratios), however feature matrix profile based measure was not used, which has distinct advantage when it comes finding similar features for voice profile recognition. Bhattacharyya histogram is used to measure the distance between the histogram profiles of the feature matrix made of audio signals (Hindi Phenomes).

KEYWORDS: MFCC, K-Means Algorithm, Framing, Windowing, Hamming Window, Fast Fourier Transform, Mel-Scaled Filter Bank, Bhattacharyya Coefficient, ROC Curve

INTRODUCTION

Speech recognition is the process by which a algorithm identifies spoken words. Basically, it means talking to your algorithms and having it correctly recognize what one is saying in simple words. However, the basic terms for understanding the basic are: Utterance, Speaker Dependence, Vocabularies. The term phoneme was reportedly first used by A. Dufriche-Desgenettes in 1873, but it referred only to a speech sound. The term phoneme as an abstraction was developed by the Polish linguist Jan Niecisław Baudouin de Courtenay and his student Mikołaj Kruszewski during 1875–1895[1]. The term used by these two was fonema, the basic unit of what they called psychophonetics. The concept of the phoneme was then elaborated in the works of Nikolai Trubetzkoi and others of the Prague School (during the years 1926–1935), and in those of structuralists like Ferdinand de Saussure, Edward Sapir, and Leonard Bloomfield. Some structuralists (though not Sapir) rejected the idea of a cognitive or psycholinguistic function for the phoneme[2][3].

Units of Speech

A phoneme is a basic unit of a language's phonology, which is combined with other phonemes to form meaningful units such as words or morphemes. The phoneme can be described as "he smallest contrastive linguistic unit which may bring about a change of meaning". [6] In this way the difference in meaning between the English words kill and kiss is a result of the exchange of the phoneme /l/ for the phoneme /s/. Two words that differ in meaning through a contrast of a single phoneme are called minimal pairs. Some linguists (such as Roman Jakobson and Morris Halle) proposed that phonemes may be further decomposable into features, such features being the true minimal constituents of language.[7]

Features overlap each other in time, as do supra segmental phonemes in oral language and many phonemes in sign languages. Features could be characterized in different ways: Jakobson and colleagues defined them in acoustic terms, [8] Chomsky and Halle used a predominantly articulatory basis, though retaining some acoustic features, while Ladefoged's system[9] is a purely articulatory system apart from the use of the acoustic term 'sibilant'. By analogy with the phoneme, linguists have proposed other sorts of underlying objects, giving them names with the suffix -eme, such as morpheme and grapheme. These are sometimes called emic units. The latter term was first used by Kenneth Pike, who also generalized the concepts of emic and etic description (from phonemic and phonetic respectively) to applications outside linguistics[10]

K-MEANS ALGORITHM

The K-means algorithm partitions the T feature vectors into M centroids. The algorithm first randomly chooses M cluster-centroids among the T feature vectors. Then each feature vector is assigned to the nearest centroid, and the new centroids are calculated for the new clustres. This procedure is continued until a stopping criterion is met, that is the mean square error between the feature vectors and the cluster-centroids is below a certain threshold or there is no more change in the cluster-center assignment.

In other words, the objective of the K-means is to minimize total intra-cluster variance,

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2 \quad (1)$$

where there are *k* clusters *Si*, *i*= 1,2,...,*k* and μ_i is the centroid or mean point of all the points.

MFCC

MFCC is mel frequency cpestral coefficients (MFCC). It is one of the most important features, which is required among various kinds of speech applications. It shows high accuracy results for clean speech and can be regarded as the "standard" features in speaker as well as speech recognition. However, experiments show that the parameterization of the MFC coefficients which is best for discriminating speakers is different from the one usually used for speech recognition applications.

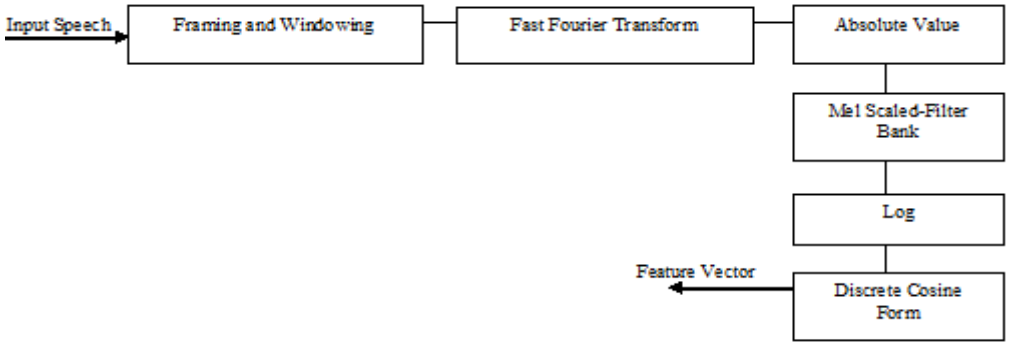

Figure 1: MFCC Flow Diagram

Framing

To enhance the accuracy and efficiency of the extraction processes, speech signals are normally framed before features are extracted. Speech signal framing covers digital filtering and speech signal detection. Framing includes pre-emphasis filter and filtering out any surrounding noise using several algorithms of digital filtering.

Windowimg

The multiplication of the speech wave by the window function has two effects :-

- It gradually attenuates the amplitude at both ends of extraction interval to prevent an abrupt change at the endpoints.
- It produces the convolution for the Fourier transform of the window function and the speech spectrum.

There are many types of windows such as : Rectangular window, Hamming window, Hann window, Cosine window, Lanczos window, Bartlett window (zero valued end-points), Triangular window (non-zero end-points), Gauss windows.

we used hamming window the most common one that being used in speaker recognition system.

The hamming window $W_H(n)$, defined as [3] :-

$$W_H(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

The use for hamming windows is due to the fact that mfcc will be used which involves the frequency domain (hamming windows will decrease the possibility of high frequency components in each frame due to such abrupt slicing of the signal).

Fast Fourier Transform

The basis of performing fourier transform is to convert the convolution of the glottal pulse and the vocal tract impulse response in the time domain into multiplication in the frequency domain [3][4]. Spectral analysis shows that different timbres in speech signals corresponds to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame. When we perform FFT on a frame, we assume that the signal within a frame is periodic, and continuous when wrapping around. If this is not the case, we can still perform FFT but the in continuity at the frame's first and last points is likely to introduce undesirable effects in the frequency response. To deal with this problem, we have two strategies:

- Multiply each frame by a Hamming window to increase its continuity at the first and last points.
- Take a frame of a variable size such that it always contains an integer multiple number of the fundamental periods of the speech signal.

Mel-Scaled Filter Bank

The speech signal consists of tones with different frequencies. For each tone with an actual Frequency, f, measured in Hz, a subjective pitch is measured on the 'Mel' scale. The mel-frequency scale is a linear frequency spacing below 1000Hz and a logarithmic spacing above 1000Hz. We can use the following formula to compute the mels for a given frequency *f* in Hz:

$$mel(f) = 2595*\log10(1+f/700) \text{ [4].} \quad (3)$$

Mel filterbanks are non-uniformly spaced on the frequency axis, so we have more filters in the low frequency regions and less number of filters in high frequency regions [5].

FEATURE MATCHING

In the recognition phase an unknown speaker, represented by a sequence of feature vectors {x1,x2,….,xT}, is compared with the codebooks in the database. For each codebook a distortion measure is computed, and the speaker with the lowest distortion is chosen,

$$C_{best} = \underset{1 \le i \le N}{argmin} \{s(X, C_i)\} \quad (4)$$

One way to define the distortion measure, which is the sum of squared distances between vector and its representative (centroid), is to use the average of the Euclidean distances:

$$s(X, C_i) = \frac{1}{T} \sum_{t=1}^{T} d\left(x_t, c_{min}^{i,t}\right) \quad (5)$$

The well known distance measures are Euclidean, city distance, weighted Euclidean and Mahalanobis. But we will not use Euclidean distance. Instead Bhattacharyya distance is used. Which measures the similarity of two discrete or continuous probability distributions as in our case audio signal features with their histograms. It is closely related to the Bhattacharyya coefficient which is a measure of the amount of overlap between two statistical samples.

Table 1: Recognition Results of Hindi Phenomes

Word	Recognition Rate(%)
0(shoonya)	100
1(ek)	79
2(do)	79
3(teen)	81.25
4(char)	68.75
5(paanch)	68.75
6(chae)	54
7(saat)	84.25
8(aath)	84.25
9(nau)	79
average	97.65

The recordings of speakers were done. The recognition vocabulary consisted of Hindi Digits (0, pronounced as “shoonya” to 9, pronounced as “nau”). Much of the error in recognition can be attributed to the presence of plosives at the beginning or end of some of the words, as in, “paach” (begins and ends in a plosive), and, “ek” (ends in a plosive), which are misinterpreted as more than one word. Further, the Hindi digit vocabulary is a confusing vocabulary. The digits 4 (pronounced as “char”), 7 (pronounced as “saat”) and 8 (pronounced as “aath”), have the same vowel part, and differ only in their unvoiced beginnings and endings. Similarly, the digits 2 (pronounced as “do”), and 9 (pronounced as “nau”) have a very similar vowel part, and differ in their beginnings. The digits 0 (pronounced as “shoonya”) and 3 (pronounced as “teen”) which are very distinct from the rest of the digits are seen to have a very high recognition rate.

BHATTACHARYYA COEFFICIENT

The **Bhattacharyya coefficient** is an approximate measurement of the amount of overlap between two statistical samples. The coefficient can be used to determine the relative closeness of the two samples being considered.

Calculating the Bhattacharyya coefficient involves a rudimentary form of integration of the overlap of the two samples. The interval of the values of the two samples is split into a chosen number of partitions and the number of members of each sample in each partition is used in the following formula,

$$\text{Bhattacharyya} = \sum_{i=1}^{n} \sqrt{(\sum a_i \cdot \sum b_i)} \quad (6)$$

where considering the samples **a** and **b**, **n** is the number of partitions, and $\sum a_i$, $\sum b_i$ are the number of members of samples **a** and **b** in the **i**'th partition.

The Bhattacharyya coefficient will be 0 if there is no overlap at all due to the multiplication by zero in every partition. This means the distance between fully separated samples will not be exposed by this coefficient alone.

ROC CURVE

ROC is Receiving Operating Characteristics which provides a useful means to assess the diagnostic accuracy of a Hindi Phenome Recognition and to compare the performance of more than one Hindi Phenome Recognition for the same Outcome.

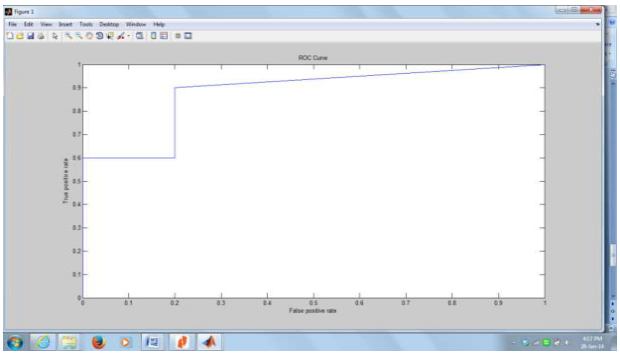

Figure 2: ROC Curve (False Positive Rate/True Positive Rate)

A perfect Hindi Phenome Recognition would have sensitivity and specificity both equal to 1. If a cut-off value existed to produce such a Hindi Phenome Recognition, then the sensitivity would be 1 for any non-zero values of 1 – specificity. The ROC curve would start at the origin (0,0), go vertically up the y-axis to (0,1) and then horizontally across to (1,1). A good Hindi Phenome Recognition would be somewhere close to this ideal.

If a variable has no diagnostic capability, then a Hindi Phenome Recognition based on that variable would be equally likely to produce a false positive or a true positive:

Sensitivity = 1 – specificity, or

Sensitivity + specificity = 1

The position of the ROC on the graph reflects the accuracy of the diagnostic test. It covers all possible thresholds (cut-off points). The ROC of random guessing lies on the diagonal line. The ROC of a perfect diagnostic technique is a point at the upper left corner of the graph, where the TP proportion is 1.0 and the FP proportion is 0.

CONCLUSIONS

The goal of this work was to implement a unit of hind speech with respect to the speaker. We have found that the system is an efficient and simple way to do speaker identification with unit of hindi speech spoken. Our system is 97.65% accurate in identifying the correct speaker when using 30 seconds for training session and several ten seconds long for testing session.

REFERENCES

1. Singhvi et al. (2008). "Hierarchical phoneme classifier for Hindi speech," Signal Processing, 2008. ICSP 2008. 9th International Conference on, vol., no., pp. 571, 574, 26-29 Oct. 2008.
2. Hailemariam et al. (2007). "Extraction of Linguistic Information with the AID of Acoustic Data to Build Speech Systems," Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol.4, no., pp. IV- 717, IV-720, 15-20 April 2007.
3. Chihi et al. (2012). "Recurrent Neural Network learning by adaptive genetic operators: Case study: Phonemes recognition," Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on, vol., no., pp.832, 834, 21-24 March 2012.
4. Heracleous et al. (2012). "Continuous phoneme recognition in Cued Speech for French," Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, vol., no., pp. 2090, 2093, 27-31 Aug. 2012.
5. Kumar, M et al. (2004). "A large-vocabulary continuous speech recognition system for Hindi," IBM Journal of Research and Development, vol. 48, no. 5.6, pp.703, 715, Sep. 2004.
6. Gray, A et al. (1976). "Distance measures for speech processing," Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 24, no. 5, pp. 380, 391, Oct 1976.
7. Cutajar, M et al (2013). "Discrete wavelet transforms with multiclass SVM for phoneme recognition," EUROCON, 2013 IEEE, vol., no., pp.1695,1700, 1-4 July 2013.
8. Sharifzadeh, S et al. (2012). "Spectro-temporal analysis of speech for Spanish phoneme recognition," Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on, vol., no., pp.548, 551, 11-13 April 2012.
9. Kshirsagar, A et al. (2012). "Comparative Study of Phoneme Recognition Techniques," Computer and Communication Technology (ICCCT), 2012 Third International Conference on, vol., no., pp. 98, 103, 23-25 Nov. 2012.
10. Cutajar, M et al. (2012). "Comparison of different multiclass SVM methods for speaker independent phoneme recognition," Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on, vol., no, pp.1, 5, 2-4 May 2012.